Spinnaker - Problems and Solutions

State of Spinnaker:

- Spinnaker is currently installed and deployed on the cluster (for full steps see this document)

Problems:

- When trying to expose the UI (spin-deck), we are running into issues. These issues seem to stem from the UI being unable to communicate with the API (spin-gate). After exposure, the UI is briefly reachable, but will quickly begin to timeout when trying to visit the URL of the ELB fronting the UI.
- Deploy stage does not deploy
  - Need a service for Blue/Green Deployment, but the service cannot have the any

More Details on Solutions Attempted:

- Created a LoadBalancer service for both UI (spin-deck) and API (spin-gate), per the following documentation: https://kb.armory.io/admin/expose-spinnaker/
  - After exposure via this method, the UI is briefly reachable. That is, we are able to visit the URL of the ELB, but upon inspection in the browser console we see a stream of errors. Shortly after, visiting the same URL will result in Timeout Errors.
    - For Jordan, the first error is always:
      - Access to XMLHttpRequest at '<elb-ip-for-spin-gate>:8084/credentials?expand=true' from origin '<elb-ip-for-spin-deck>:9000/' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'. The credentials mode of requests initiated by the XMLHttpRequest is controlled by the withCredentials attribute.
    - Bafflingly, other members of the team see different errors initially. (Jordan also sees these errors, just later)
      - https://i.imgur.com/ggEqq28.png
      - https://cdn.discordapp.com/attachments/715652345878544485/718573061867044867/unknown.png
- Added expected CORS pattern via two methods (which effectively do the same thing). This has not yielded any different results.
  - hal config security api edit --cors-access-pattern <elb-ip-for-spin-deck>
  - Added pattern into config file manually per this documentation
- Created “proper” DNS addresses for ui and api (instead of using ELB address). Followed method similar to the one on this doc starting at the paragraph that begins with “Now authorize the UI and API servers...”

    - This resulted in an alleviation of the timeout issue for UI. UI is now consistently reachable, but is still not talking to API.

Other Interesting Things to Note:

- Running **kubectl logs** on the pods hosting spin-gate and spin-deck do not reflect any errors. HTTP Status 200 all the way down
- CURLing the API returns the expected info

Ideas for Tracking Issue:

- Deploy Spinnaker on a personal GCP cluster to see if the issue persists.
    - Pros:
        - See if this is a cloud agnostic issue
            - Could be an AWS problem, or a Revature AWS problem
        - If we can get a working UI, we could quickly and easily create a pipeline and export that as JSON
        - Potentially get insight into issue
    - Cons:
        - Time and resource consuming, but it sounds like there will be an influx of manpower to the Spinnaker team next week.
        - Could hit the same issue and end up wasting a bunch of time

Ideas for problem mitigation:

- Use spin cli commands and a pipeline in JSON format to add and execute pipelines with Spinnaker.
- We are currently sending traffic to the local ports on the pod and forwarding that traffic to our local machines. This work around currently works, but is not very stable or extensible. Efforts still need to be made to expose the ui on the DNS address.
- Add in authentication (probably OAuth) before entering deck DNS. Hopefully this would help mitigate the CORS issue (The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'. The credentials mode of requests initiated by the XMLHttpRequest is controlled by the withCredentials attribute.)